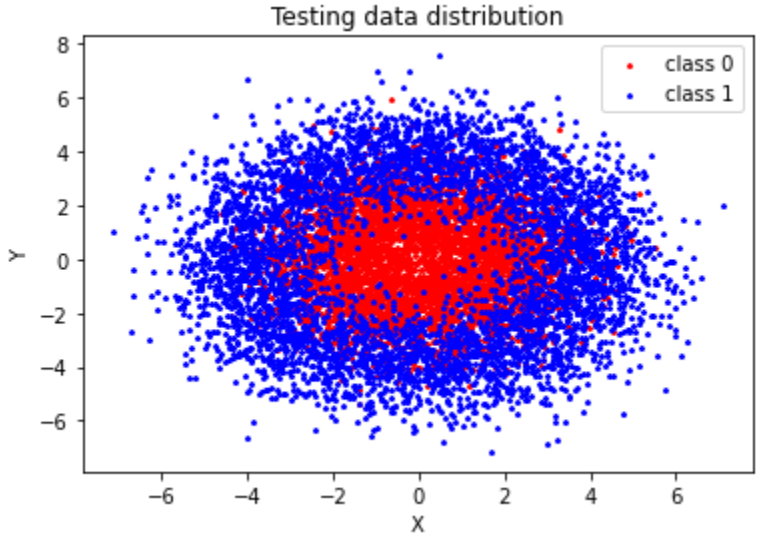
EECE 5644 – Goutham Saravanan – Project Alternative

Q1.

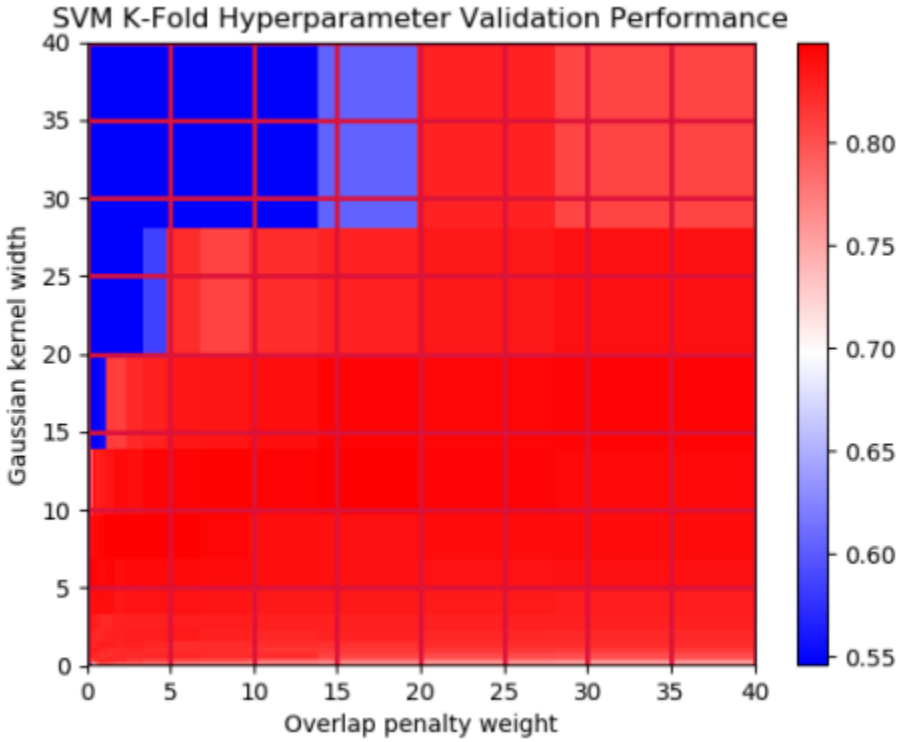
The class priors were **0.5** and **0.5** for class 0 and class 1 respectively. Plots generated for the above mentioned priors train and test datasets are shown below.

1000 independent and identically distributed (iid) samples were generated for training and 10000 iid samples for testing.



SVM Classification:

During SVM K-Fold validation, 20 values of the box constraint parameter(C) and Gaussian kernel width(K) in the ranges [0.5, 40] and [0.5, 40] were tested, respectively. The accuracy of a model for each tested combination is depicted in the heatmap below.

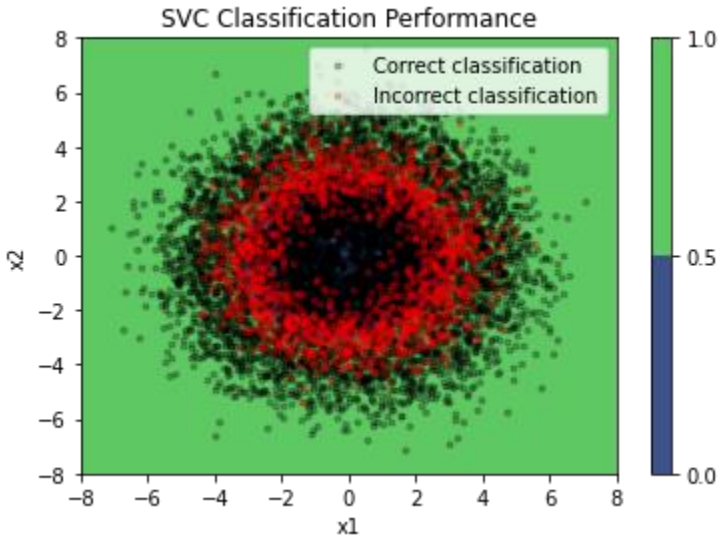


When the hyperparameters C and K were **1.185** and **6.87**, the greatest accuracy (**0.86**) was reached on the training dataset using 10-fold cross-validation.

Gamma was calculated as Gamma = 1/(2 * kernel width**2)

from the gaussian kernel width. 0.01 is the best gamma value. The Support Vector Classifier was trained on the whole training dataset and verified on the test dataset including 10,000 samples using these hyperparameters. On the test dataset, the accuracy was **0.834**.

The Support Vector Classifier was trained on 1000 samples and evaluated on 10,000 samples, as seen in the graph below. Because both classes' data distributions are circular, the decision boundary dividing them is circular as well. The decision border was created by multiplying the confidence threshold by 0.5 and superimposing it on the figure below. Correct and erroneous categorization are shown by black and red dots, respectively. The samples that are classed as class 0 and class 1 are colored blue and green, respectively. As predicted, there are more misclassifications in the decision border since there is some overlap between two classes.



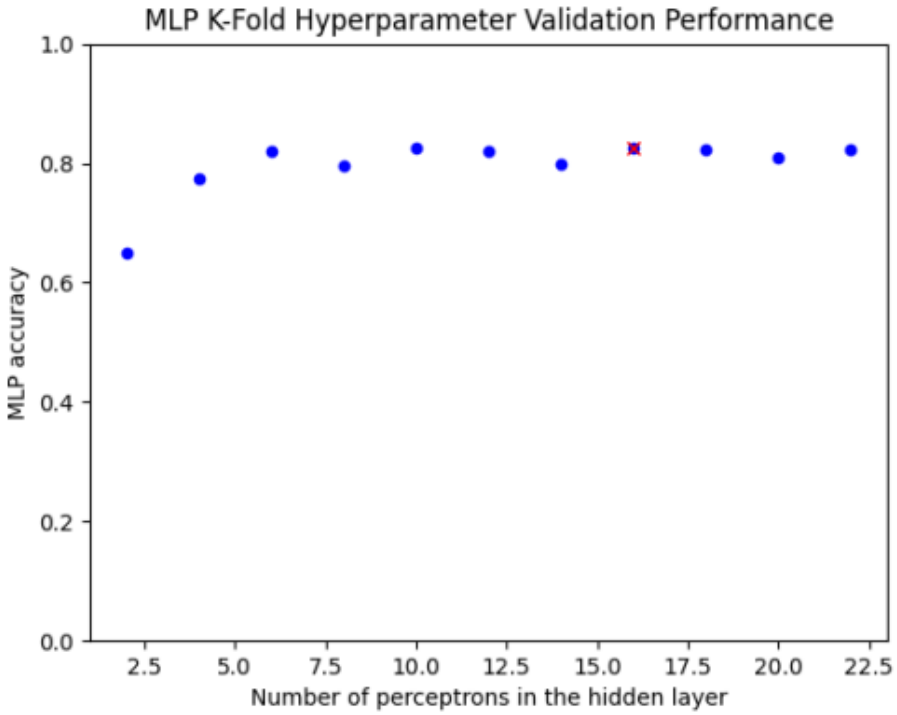
Multi-Layer Perceptron :

I used a two-layer MLP with one hidden layer and one output layer for this segment. The activation function for the buried layer was RELU. Because there are just two classes, sigmoid was chosen as the output layer's activation function. Binary cross-entropy loss was employed as the loss function, while Stochastic Gradient descent was used as the optimizer. For 300 epochs, the model was trained.

Calculating accuracy attained on 10-fold cross-validation for a range of values was used to determine the ideal number of perceptrons for the hidden layer. The range of values evaluated to discover an ideal number of perceptrons for this segment was 2 to 22.

On the training dataset, the ideal number of perceptrons for achieving maximum accuracy using 10-fold cross-validation was 16. On the training dataset, the accuracy was **0.832**.

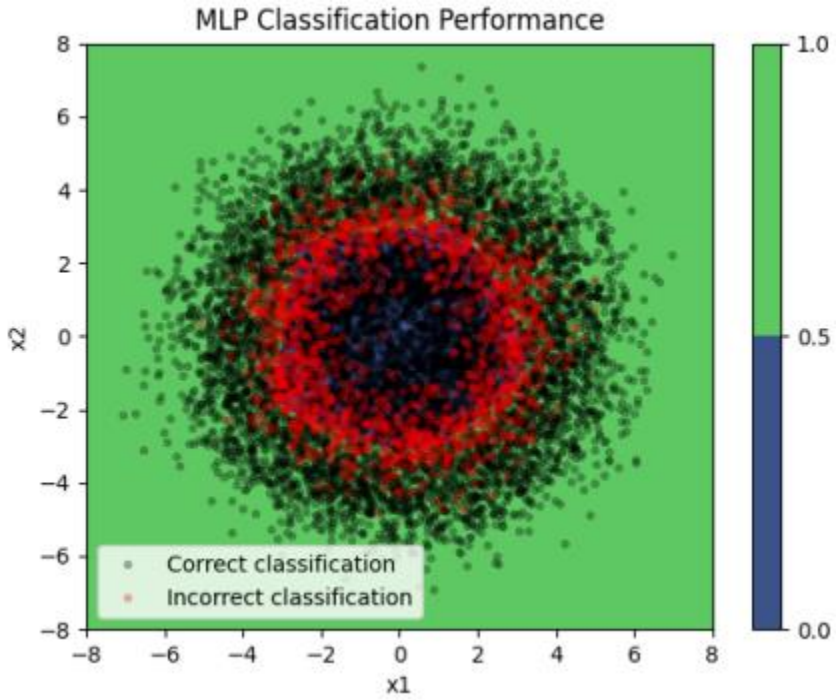
The accuracy of the MLP model for a different number of perceptrons in the hidden layer is shown in the graph below.



The architecture of the final MLP with the optimal number of perceptrons used to train on the whole training dataset is shown in the image below. The test dataset was utilized to validate the final model. On the test dataset, the accuracy is **0.845**.

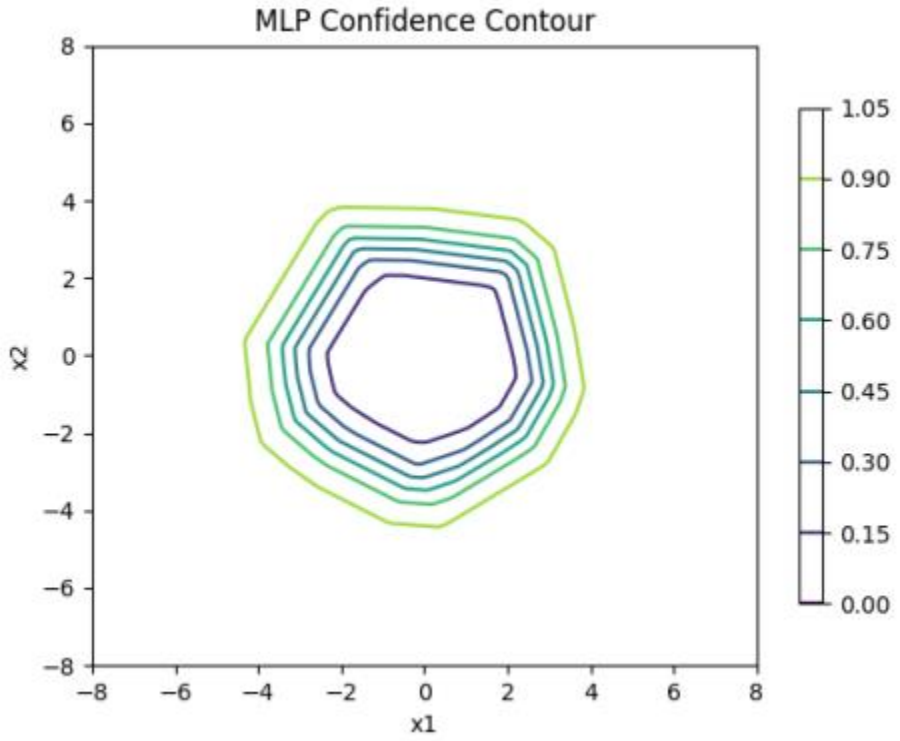
```
  "The `lr` argument is deprecated, use `learning_rate` instead.")
model summary
Model: "sequential_110"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_220 (Dense)            (100, 18)                 54
_________________________________________________________________
dense_221 (Dense)            (100, 1)                  19
=================================================================
Total params: 73
Trainable params: 73
Non-trainable params: 0
_________________________________________________________________
```

The MLP Classifier was trained on 1000 samples and evaluated on 10,000 samples, as seen in the graph below. Both classes were classified using a 0.5 threshold. Correct and erroneous categorization are shown by black and red dots, respectively.



The samples that are classed as class 0 and class 1 are colored blue and green, respectively. As predicted, there are more misclassifications in the decision border since there is some overlap between two classes.

The confidence contour of the MLP model on the test dataset is shown below for various confidence levels.



	Train Accuracy(10-fold) /1000 samples	Train Accuracy(10-fold) /10000 samples
SVM	0.86	0.834
MLP	0.832	0.845

When compared to the MLP classifier, SVM has a greater training accuracy. However, when compared to the SVM classifier, MLP has a greater classification accuracy on the test dataset. Because the test dataset contains a large number of data samples, there are numerous samples in the boundary between two classes. Because the SVM classifier is sensitive to data samples at the border, increasing the number of data samples decreased accuracy while validating on the test dataset. In the case of the MLP classifier, this is not the case. On the test database, the accuracy improved.

Q2.

Using the row index, column index, red, green, and blue pixel values, a 5-dimensional feature vector was constructed from the input image for this question. Then, using maximum likelihood parameter estimation, a Gaussian Mixture Model was fitted. Using maximum average validation-log-likelihood as the goal function and 10-fold cross validation, the ideal number of Gaussian components was calculated.

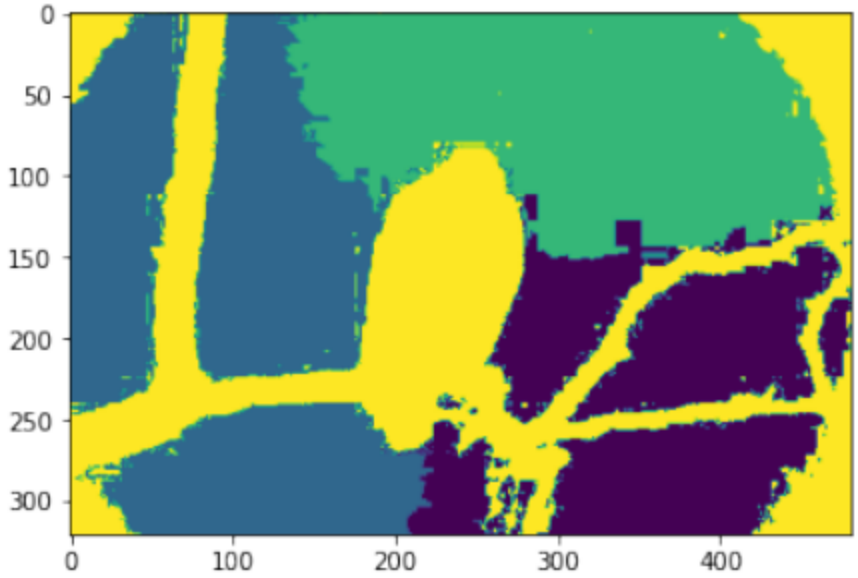
All of the RGB photos used in this article are dimensioned (321, 481). The total amount of pixels is 321 X 481 pixels, or 1,544,401 pixels, with each pixel containing three R, G, and B values.

The normalize function from the sklearn library was used to normalize each of the 5 features to a [0,1] scale. The GaussianMixture() method from the scipy package was used to fit the GMM.

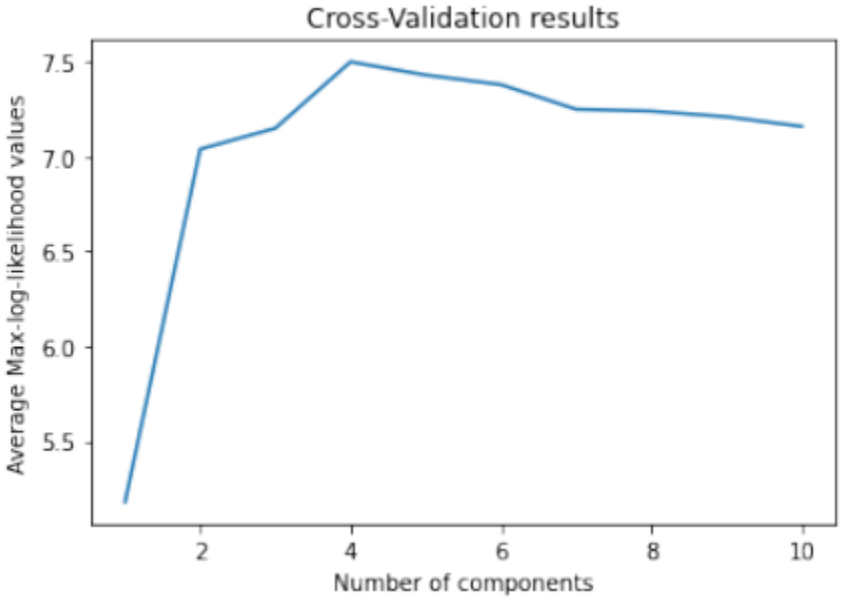
For each cluster, the pdf was calculated for the normalized feature vector, and the cluster with the highest probability was mapped to that pixel.

Three RGB photos and their pixel-by-pixel segmentation using GMM are shown below. The best number of Gaussian components was calculated for all of the images below by finding the maximum average validation-log-likelihood for a range of values ranging from 1 to 11.

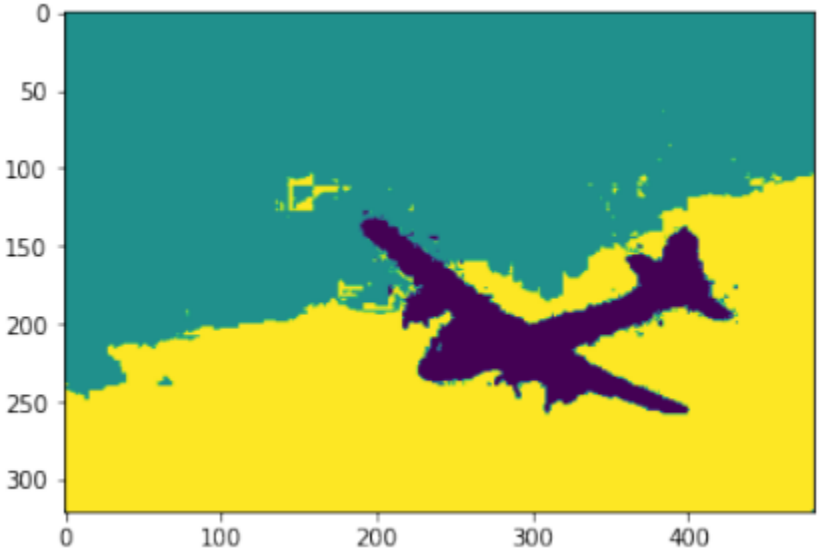
1st image :



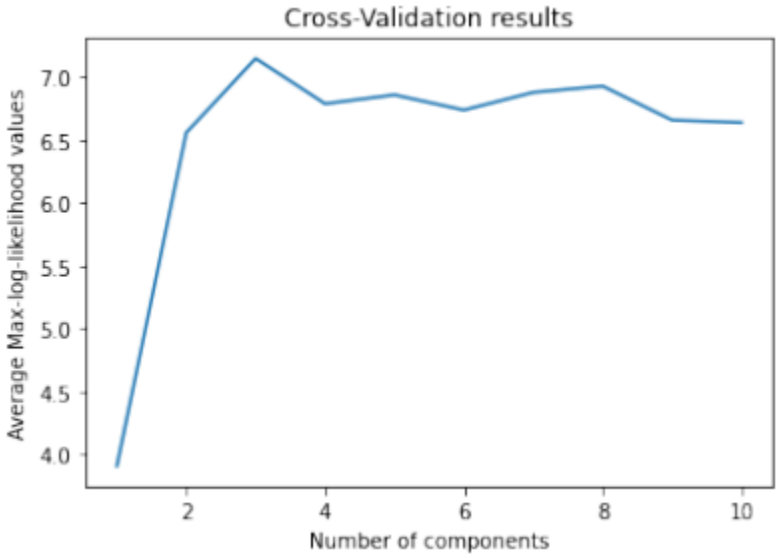
The average validation-log-likelihood for a variety of gaussian components is shown in the graph below.
Optimal number of gaussian components for this image is 4.



2nd image :



The average validation-log-likelihood for a variety of gaussian components is shown in the graph below.
Optimal number of gaussian components for this image is 3.



APPENDIX (CODE) GDrive Links :

Q1: bit.ly/gouthammlhw5q1

Q2: bit.ly/gouthammlhw5q2